

# Hibridación de dos Algoritmos Evolutivos para la Optimización de Funciones Multiobjetivo: MOPSO y ESN

Sandra García Rodríguez<sup>1</sup>, Inés M. Galván León<sup>2</sup>

**Resumen**—El presente trabajo de investigación tiene como objetivo estudiar la hibridación de dos algoritmos multiobjetivo: Enjambres de partículas (MOPSO) y un algoritmo multiobjetivo basado en la combinación de NSGA-II con Estrategias Evolutivas (ESN). Se pretende analizar si la hibridación permite obtener frentes de Pareto mejores que los obtenidos individualmente por los algoritmos ya que, en estudios previos sobre estos algoritmos, se observó que, para ciertos problemas, un algoritmo puede ayudar a otro (y viceversa) en la obtención de frentes más óptimos. Una forma de plantear esta hibridación es utilizar la población obtenida por un algoritmo para inicializar el otro y, para ello, se han realizado experimentos ejecutados de manera homogénea, para cada una de las aproximaciones así como para la hibridación de ambas, con cuatro funciones teóricas (ZDT1, ZDT2, ZDT3 y ZDT4) y un problema real: MANETs.

## I. INTRODUCCIÓN

Hoy en día, muchos de los problemas de optimización que se presentan en el mundo real son, naturalmente, multiobjetivo, lo que quiere decir que suelen tener dos o más funciones objetivo que deben satisfacerse simultáneamente y que puede ocurrir que estén en conflicto entre sí. De existir la posibilidad de combinar los diferentes objetivos y conociendo la mejor manera de hacerlo, se puede simplemente considerar la existencia de un único objetivo a optimizar, resultante de la combinación de todos los objetivos considerados. Sin embargo, lo usual es que no se conozca de que manera combinarlos, o esto sea inadecuado, cuando no imposible. Entonces, se trata de un Problema de Optimización Multiobjetivo (*Multiobjective Optimization Problem*).

Un **Problema de Optimización Multiobjetivo** (MOP) general incluye un conjunto de  $n$  parámetros (o variables de decisión), un conjunto de  $k$  funciones objetivo, y un conjunto de  $m$  restricciones. Las funciones objetivo y las restricciones son funciones de las variables de decisión. Por lo que un MOP puede expresarse como:

$$\begin{array}{ll} \text{Optimizar} & \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \\ \text{sujeto a} & \mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), \dots, e_m(\mathbf{x})) \geq \mathbf{0} \\ \text{donde} & \mathbf{x} = (x_1, x_2, \dots, x_n) \in X \text{ e} \\ & \mathbf{y} = (y_1, y_2, \dots, y_k) \in Y \end{array}$$

siendo  $\mathbf{x}$  el vector de decisión e  $\mathbf{y}$  el vector objetivo.

En los MOP con objetivos contradictorios no existe una solución única que pueda ser considerada la mejor, sino un conjunto de alternativas que representan los mejores compromisos entre todos los objetivos, en el sentido que cada solución es mejor que las otras en algún objetivo, pero ninguna es mejor que otra en todos los objetivos simultáneamente. A dicho conjunto se le denomina conjunto de soluciones “Óptimo de Pareto” y sus correspondientes vectores en el espacio de objetivos constituyen el denominado “Frente de Pareto” [7].

Los Algoritmos Evolutivos (*MultiObjective Evolutionary Algorithms* o *MOEAs*) han demostrado ser especialmente adecuados para la optimización multiobjetivo. Algunos de los más conocidos son SPEA (*Strength Pareto Evolutionary Algorithm*), SPEA2 [16], PAES [8] o NSGA-II [5].

En los últimos años, la hibridación de metaheurísticas ha recibido un considerable interés, proponiéndose una amplia variedad de de aproximaciones basadas en la hibridación de algoritmos, de las cuales se han obtenido muy buenos resultados, como por ejemplo la combinación de algoritmos “búsqueda de descenso local”, “temple simulado”, “búsqueda tabu” o “algoritmos evolutivos”. Además, existen artículos en los que se estudian más a fondo estos trabajos pudiendo clasificar dichas hibridaciones [13]. También, se han desarrollado varias herramientas que permiten este tipo de experimentos, cabe citar como ejemplo PISA [2], Paradiseo [14] o Team [10].

El objetivo de este trabajo es validar la hibridación de dos algoritmos evolutivos. Éstos son MOPSO [3], el cual permite la optimización mediante enjambres de partículas, y un algoritmo multiobjetivo (ESN) [12] basado en la combinación de NSGA-II con Estrategias Evolutivas, el cual emplea los pasos estándar las Estrategias Evolutivas solo que reemplazando su proceso de selección de individuos por el proceso de selección de NSGA-II. Debido a que, en estudios previos, se ha observado que para ciertos problemas un algoritmo puede ayudar a otro (y viceversa) en la obtención de frentes más óptimos, se quiere analizar si, gracias a esta hibridación, se pueden conseguir frentes de Pareto mejores que los obtenidos en las ejecuciones

<sup>1</sup> Dirección del autor 1. E-mail: sgrodrig@inf.uc3m.es

<sup>2</sup> Dirección del autor 2. E-mail: igalvan@inf.uc3m.es

de cada algoritmo por separado. Esta hibridación consiste en utilizar la población obtenida por un algoritmo para inicializar el otro. Se ha empleado en experimentos con problemas teóricos (como las funciones ZDT1, ZDT2, ZDT3 y ZDT4) [15] y para un problema real conocido como MANETs.

## II. ALGORITMOS MULTI-OBJETIVO

Como se ha comentado en la introducción, el objetivo es estudiar y validar la hibridación de los algoritmos MOPSO y ESN. En este artículo se plantea la hibridación de tal manera que la población obtenida por un algoritmo se utiliza para inicializar el otro, es decir, comienza la ejecución de un algoritmo durante un número determinado de ciclos para después dar paso al segundo algoritmo el cual recibe como población inicial la salida del primero.

Cada individuo representa, para MOPSO, una partícula con su posición, velocidad y otros parámetros asociados, mientras que en ESN se trabaja con individuos con varianza propia. Debido a esto, sólo serán de utilidad las posiciones de los miembros de la población que se transmitan de un algoritmo a otro. Es decir, ESN necesita una varianza inicial que MOPSO no utiliza, por lo que cuando se ejecuten pruebas en las que comience por MOPSO, se añadirá una varianza nueva a la población de salida con la que se inicia ESN. En caso de comenzar la ejecución por ESN, a la población que recibe MOPSO se le asignará una velocidad inicializándose, además, el resto de parámetros del individuo necesarios para el algoritmo.

A continuación, se describen con mayor detalle los algoritmos evolutivos utilizados en este trabajo: MOPSO y ESN.

### MOPSO

MOPSO (*Multiobjective Particle Swarn Optimization*) fue propuesto por Carlos A. Coello Coello [4]. Este algoritmo permite la optimización mediante enjambres de partículas con el fin de resolver problemas que tengan varias funciones objetivo. En él la población está compuesta por partículas, cada una tiene asociados una posición y una velocidad entre otros parámetros.

A continuación se procede a explicar el funcionamiento del algoritmo:

- 1) Inicializar la población.
- 2) Calcular el fitness para cada partícula y guardarlo, junto con su posición, en *PBestFitness* y *PBestPosition*.
- 3) Guardar la posición y el fitness de las partículas no dominadas en el repositorio.
- 4) Mientras no se alcance el número máximo de ciclos, para cada partícula hacer:
  - a. Seleccionar un líder del repositorio.

- b. Actualizar la velocidad de la partícula (*NewVelocity*) usando las ecuaciones estándar y el líder como vecino.
- c. Actualizar la posición de la partícula (*NewPosition*) usando las ecuaciones estándar de PSO.
- d. Calcular el nuevo fitness (*NewFitness*).
- e. Si el *NewFitness* domina *PBestFitness* entonces:
  - i. Actualizar *PBestFitness* con el nuevo fitness y *PBestPosition* con la nueva posición.
  - ii. Si *NewFitness* es no dominado por las soluciones del repositorio entonces:
    1. Insertar *NewFitness* y *NewPosition* en el repositorio.
    2. Eliminar del repositorio aquellas soluciones dominadas por el recién llegado.

### ESN

Está basado en el uso combinado entre el algoritmo de Estrategias Evolutivas y el algoritmo de NSGAI (*Nondominated Sorting Genetic Algorithm* versión 2). Este híbrido emplea los pasos estándar de ESN solo que reemplazando su proceso de selección de individuos por el proceso de selección de NSGAI. La población estará compuesta por individuos con una varianza propia, la cual indicará la magnitud del cambio que se produzca en el individuo cuando éste vaya a ser mutado.

El esquema del algoritmo es el siguiente:

- 1) Inicializar la población  $P$  de  $\mu$  individuos.
- 2) Inicializar la varianza  $\sigma$  de cada individuo  $I \in P$ .
- 3) Mientras no se alcance el número máximo de ciclos:
  - a. Se inicializa  $P'$  como vacía.
  - b. Para cada individuo  $I = (x_1', \dots, x_n', \sigma) \in P$  hacer:
    - i.  $\sigma' = \sigma * e^{N(0, \text{RAND\_MAX})}$ .
    - ii. Crear el nuevo individuo  $I' = (N(x_1, \sigma'), N(x_2, \sigma'), \dots, N(x_n, \sigma'), \sigma')$ .
    - iii. Añadir  $I'$  a la población  $P'$ .
  - c. Añadir a  $P$  la población  $P'$ .
  - d. Calcular el frente  $F_1$  con individuos no dominados de  $P$ .
  - e. Desde la iteración  $i=2$  hasta el máximo de ciclos:
    - i. Generar los frentes  $F_i$  con individuos no dominados de  $P \setminus (F_1 \cup F_2 \cup \dots \cup F_{i-1})$ .
    - ii. Ordenar las soluciones en  $F_i$  usando la distancia de crowding.
    - iii. Borrar los últimos  $\mu$  individuos de la población  $P$ .

### III. DOMINIOS DE EXPERIMENTACIÓN

#### A. Funciones Teóricas

Los problemas ZDT [15] constituyen un conjunto de seis problemas, de los cuales se escogen cuatro, que contemplan distintas posibilidades y dificultades al considerar problemas multiobjetivo. Las cuatro funciones elegidas (ZDT1, ZDT2, ZDT3 y ZDT4) representan problemas de minimización con dos objetivos. Las tres primeras dependen de 30 variables en el rango  $[0,1]$  ( $x = (x_1, \dots, x_n)$ ), mientras que la cuarta tiene 10 variables en rangos diferentes  $x_1 \in [0,1]$ ,  $x_2, \dots, x_9 \in [-5,5]$ . Estas funciones se definen de la siguiente manera:

ZDT1:

$$f_1(x) = x_1$$

$$f_2(x) = g(x) * [1 - \sqrt{\frac{x_1}{g(x)}}]$$

$$g(x) = 1 + 9 * \frac{(\sum_{i=2}^n x_i)}{n-1}$$

ZDT2:

$$f_1(x) = x_1$$

$$f_2(x) = g(x) * [1 - \left(\frac{x_1}{g(x)}\right)^2]$$

$$g(x) = 1 + 9 * \frac{(\sum_{i=2}^n x_i)}{n-1}$$

ZDT3:

$$f_1(x) = x_1$$

$$f_2(x) = g(x) * [1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} * \sin(10\pi x_1)]$$

$$g(x) = 1 + 9 * \frac{(\sum_{i=2}^n x_i)}{n-1}$$

ZDT4:

$$f_1(x) = x_1 ; f_2(x) = g(x) * [1 - \sqrt{x_1 / g(x)}]$$

$$g(x) = 1 + 10 * (n-1) + \sum_{i=2}^n [x_i^2 - 10 * \cos(4\pi x_i)]$$

#### B. Problema Real: MANETs

Se define MANET (Mobile Ad-hoc Networks) como una red de conexión inalámbrica donde cada terminal se puede comunicar directamente con otras estaciones. En los últimos años, ha aumentado el interés por este tipo de tecnología porque permite extender la cobertura y no necesita de ninguna estructura para funcionar. En este sentido, un terminal puede usar otras estaciones como routers con el fin de enviar información a otro terminal

saltando de estación en estación por el camino. Con el fin de permitir la comunicación entre estaciones que no están conectadas directamente, la red establece diferentes protocolos de encaminamiento ad-hoc que busquen las rutas adecuadas para transmitir el paquete al receptor.

Para este trabajo, se considera el problema de broadcasting como una sub-clase particular de MANETs conocido como “Metropolitan MANETs” el cual cubre desde pequeños centros comerciales hasta áreas metropolitanas. El objetivo es optimizar la transmisión de mensajes por la red, lo cual se convierte en un problema con tres objetivos a satisfacer: maximizar el número de estaciones alcanzadas minimizando el tráfico de la red y su makespan. Este problema ya ha sido abordado por otros autores utilizando técnicas de computación evolutiva, entre ellos [1] y [11].

Para este problema en concreto, la estrategia de broadcasting considerada para optimizar es DFCN (Delayed Flooding with Cumulative Neighborhood) [6], la cual define cinco parámetros para determinar los valores objetivo. Son los siguientes:

1. **minGain** es la mínima ganancia para rebroadcasting, toma valores en el intervalo  $[0.0, 1.0]$ .
2. **[lowerBoundRAD, upperBoundRAD]** define el valor de RAD o “retraso aleatorio (en milisegundos) para el rebroadcasting”. Se sitúa dentro del rango 0.0 a 10.0 milisegundos.
3. **proD** es la máxima densidad (entre 0 y 100).
4. **safeDensity** define la máxima densidad de terminales permitida por la red (de 0 a 100 dispositivos).

Estos parámetros caracterizan el espacio de búsqueda ya que se emplean para calcular el valor de las tres funciones objetivo. Los resultados vienen dados en un frente de Pareto en el cual se representa el valor de los objetivos de cada individuo.

### IV. RESULTADOS EXPERIMENTALES

Se han realizado un conjunto de experimentos para cada uno de los algoritmos y las hibridaciones planteadas. Todos ellos se ejecutan un total de 250 ciclos (o iteraciones del algoritmo), y se trabaja con una población de 100 individuos. Cada algoritmo, en cada iteración, evalúa una vez a cada individuo lo que significa para todos los experimentos de este trabajo que el número de evaluaciones será en todos los casos 25000. El valor de los parámetros propios de cada algoritmo queda descrito en la sección II.

En cuanto a las hibridaciones se refiere, se han realizado pruebas en ambos sentidos, variando el orden en el que intervienen los algoritmos, MOPSO o ESN, y variando también el número de evaluaciones para cada uno de los algoritmos, pero nunca superando el total de evaluaciones (25000).

En concreto, el número de evaluaciones total se ha repartido de la siguiente forma: 8000 evaluaciones para el primero y 17000 para el segundo; 12500 para cada uno de ellos; y, 17000 para el primero y 8000 para el segundo.

#### A. Métricas

Las soluciones resultantes de la experimentación realizada deben ser comparadas cuantitativamente, para lo cual se suelen utilizar métricas que miden el éxito de los algoritmos. En este trabajo se comparan las soluciones utilizando dos métricas comúnmente usadas: el hipervolumen y SetCoverage, las cuales se describen a continuación. Para la evaluación de estas métricas los datos han sido normalizados en el intervalo  $[0,1]$ .

1. **Hipervolumen (HV):** Representa el volumen encerrado en la unión de los puntos del conjunto. El volumen dominado por cualquier punto es calculado como el volumen del hipercubo definido para él mismo. Este valor vendrá definido en el intervalo  $[0, 1]$ .
2. **SetCoverage (SC):** La covarianza de un conjunto de puntos A frente a un conjunto de puntos B,  $SC(A, B)$ , es definida como la fracción de puntos en B que son dominados por un punto en A. Esta medida debe ser calculada en ambas direcciones, es decir, A contra B, y B contra A puesto que  $SC(A,B) \neq (1 - SC(B,A))$ .

#### B. Experimentación con funciones teóricas

Debido a que el simulador Madhoc que se emplea para MANETs requiere un alto tiempo computacional, se ha comenzado la investigación estudiando la validez del tipo de hibridación propuesta en este trabajo con cuatro funciones teóricas más sencillas realizando, para ello, diversos experimentos variando el orden de aparición de los algoritmos así como las iteraciones que ejecuta cada uno.

Como estas funciones teóricas poseen únicamente dos objetivos a optimizar, se mostrará también la representación gráfica de los frentes finales lo cual facilitará en gran medida la comparación de éstos. En algunos casos no se evalúan las métricas, pues los frentes obtenidos son claramente peores que otras soluciones.

Aunque se han realizado experimentos variando el orden en el se ejecutan los algoritmos, MOPSO o ESN, y variando también el número de evaluaciones para cada uno de los algoritmos, a continuación se muestran sólo los mejores resultados obtenidos de las posibles hibridaciones.

##### 1. ZDT1

Como se puede observar en la figura 1 el frente más dominante lo consigue MOPSO, obteniendo

además valores mayores en la evaluación de las métricas que los demás experimentos (ver Tabla I y Tabla II).

Para esta función se ha comprobado que la hibridación de MOPSO con ESN no obtiene mejores resultados que los conseguidos por MOPSO de manera individual. Según se aprecia en la Fig. 1 y en las Tablas I y II, la única hibridación que ha logrado obtener resultados cercanos al óptimo es aquella en la que se combina ESN con 8000 evaluaciones más MOPSO con 17000 consiguiendo, incluso, mejorar a ESN. Sin embargo, su frente de Pareto final sigue siendo inferior al obtenido por la ejecución homogénea de MOPSO con 25.000 evaluaciones.

En este caso, se observa que la hibridación no ayuda a obtener mejores resultados.

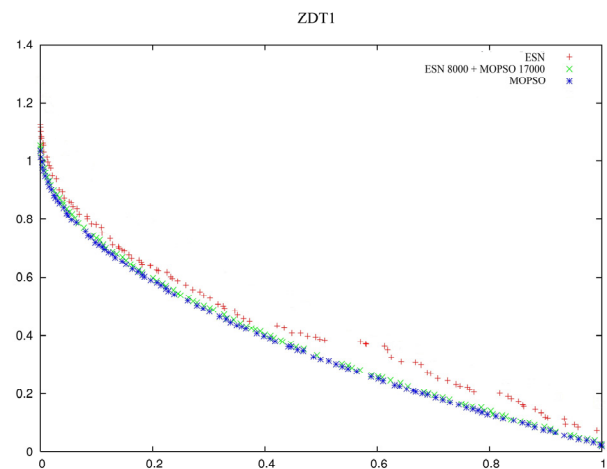


Fig. 1. Frentes ZDT1.

TABLA I  
HIPERVOLUMEN PARA ZDT1

	HV
<b>ESN-25000</b>	0.6431
<b>MOPSO-25000</b>	0.6941
<b>ESN-8000 + MOPSO-17000</b>	0.6859

TABLA II  
SETCOVERAGE PARA ZDT1

SC	ESN-25000	MOPSO-25000	ESN-8000 + MOPSO-17000
<b>ESN-25000</b>		0	0
<b>MOPSO-25000</b>	1		0.8100
<b>ESN-8000 + MOPSO-17000</b>	1	0	

## 2. ZDT2

Se ha comprobado como MOPSO no supone un algoritmo útil para resolver este problema pues no consigue encontrar, para ninguno de los experimentos realizados, más de dos puntos que compongan el frente final (ver Fig. 2), por lo que no será necesario calcular sus métricas.

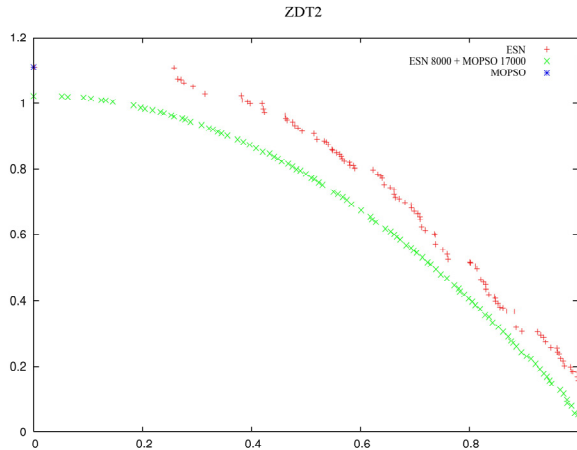


Fig. 1. Frontes ZDT2.

ESN tampoco consigue buenos resultados, los valores de HV y SC (ver Tablas III y IV) son más bajos que los obtenidos por la hibridación de ESN con 8000 evaluaciones más MOPSO con 17000. Además, si se observa la Fig. 2, el frente de ESN aparece claramente dominado por la solución conseguida por la hibridación propuesta. Además posee valores menores en la evaluación de las métricas (Tabla III y Tabla IV).

TABLA III  
HIPERVOLUMEN PARA ZDT2

	<u>HV</u>
<b>ESN-25000</b>	0.9081
<b>ESN-8000 + MOPSO-17000</b>	0.9451

TABLA IV  
SETCOVERAGE PARA ZDT2

<u>SC</u>	<b>ESN-25000</b>	<b>ESN-8000+ MOPSO-17000</b>
<b>ESN-25000</b>		0
<b>ESN-8000 + MOPSO-17000</b>	1	

Destacar que, al igual que en la función anterior, de las muchas combinaciones con las que se ha probado este tipo de hibridación, vuelve a ser mejor aquella en la que ESN comienza ejecutando 8000 evaluaciones seguidas de 17000 evaluaciones de MOPSO. Se piensa que, para este caso en concreto, ESN sirve para iniciar el algoritmo localizando individuos aptos para el frente de Pareto dando paso a MOPSO el cual se ocupa de refinarlo y mejorarlo encontrando aquellos puntos que sean más

dominantes. De esta manera, se evita que MOPSO se atasque en el inicio del algoritmo.

Con este problema se demuestra claramente que, para este dominio en concreto, el planteamiento de la hibridación de estos dos algoritmos ayuda en gran medida a mejorar los resultados obtenidos por las ejecuciones de cada uno de ellos por separado. Los resultados aquí expuestos, establecen el punto de partida de este trabajo de investigación.

## 3. ZDT3

En esta función ocurre un fenómeno parecido a la ZDT1, la mejor hibridación encontrada vuelve a ser aquella en la que se combina ESN con 8000 evaluaciones más MOPSO con 17000 evaluaciones, pero ni siquiera ésta consigue mejorar notablemente los resultados obtenidos por la ejecución de MOPSO en solitario (ver Fig. 3, Tabla V y Tabla VI) pues los valores métricos que obtienen ambos son muy similares. De igual manera, esta hibridación mejora el frente obtenido por la ejecución de ESN en solitario.

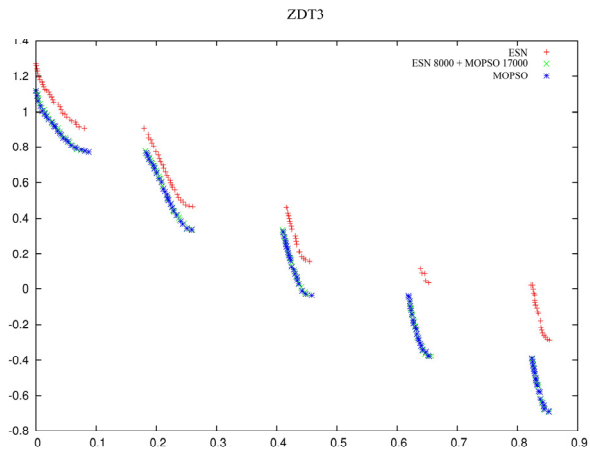


Fig. 2. Frontes ZDT3.

TABLA V  
HIPERVOLUMEN PARA ZDT3

	<u>HV</u>
<b>ESN-25000</b>	0.75
<b>MOPSO-25000</b>	0.8368
<b>ESN-8000 + MOPSO-17000</b>	0.8372

TABLA VI  
SETCOVERAGE PARA ZDT3

<u>SC</u>	<b>ESN- 25000</b>	<b>MOPSO- 25000</b>	<b>ESN-8000 + MOPSO- 17000</b>
<b>ESN-25000</b>		0	0.1268
<b>MOPSO- 25000</b>	1		0.2752
<b>ESN-8000 + MOPSO- 17000</b>	0.8130	0.2780	

#### 4. ZDT4

MOPSO tampoco resulta ser un algoritmo útil para resolver este problema ya que no encuentra los puntos necesarios en el frente final (ver Fig. 4) haciendo innecesario calcular sus métricas.

La hibridación que mayor éxito ha obtenido para las funciones anteriores, ESN con 8000 más MOPSO con 17000 evaluaciones, consigue un frente mal distribuido y extremadamente lejos del óptimo. Sin embargo, si se suben las evaluaciones a ESN, hasta llegar a 12500, restándoselas a MOPSO, para quedarse con otras 12500, se logra un frente que supera a la solución dada por la ejecución de ESN en solitario (ver Tablas VII y VIII). Se piensa que esta hibridación hace que ESN sirva para iniciar el algoritmo localizando individuos aptos para el frente de Pareto dando paso a MOPSO, cuya misión es refinarlo y mejorarlo encontrando aquellos puntos que sean más dominantes. De este modo se evita que MOPSO se atasque en comienzo del algoritmo.

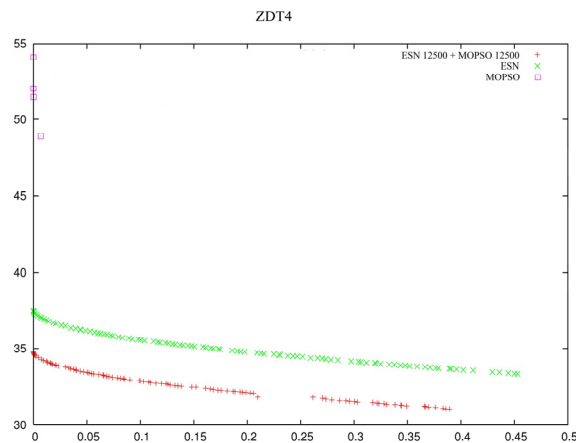


Fig. 3. Frentes ZDT4.

TABLA VII  
HIPERVOLUMEN PARA ZDT4

	HV
<b>ESN-25000</b>	0.1497
<b>MOPSO-12500 + ESN-12500</b>	0.211

TABLA VIII  
SETCOVERAGE PARA ZDT4

SC	ESN-25000	MOPSO-12500 + ESN-12500
<b>ESN-25000</b>		0
<b>MOPSO-12500 + ESN-12500</b>	1	

Como se ha observado en los apartados anteriores, el tipo de hibridación propuesto para este trabajo consigue el cometido deseado en las funciones ZDT2 y ZDT4, por lo que ya se puede empezar a pensar en que ésta técnica es plausible y capaz de conseguir mejores soluciones que las

alcanzadas hasta el momento para determinados problemas. Una vez demostrada la viabilidad de la hipótesis de partida, se probará su validez en una función real, MANETs.

#### C. Experimentación con MANETs

Para llevar a cabo esta experimentación, se emplea el simulador “Madhoc simulation” [9] que reproduce el comportamiento de MANET.

En este apartado se va a estudiar la utilidad del tipo de hibridación planteada para este trabajo para resolver este problema real. Los resultados obtenidos se muestran en las tablas IX, X, XI y XII.

Para este problema, y en cuanto a los algoritmos aislados se refiere, mencionar que ESN se comporta mejor que MOPSO en cuanto a las métricas evaluadas (ver Tablas IX y X).

Se observa que, en el caso del Hipervolumen (ver Tabla IX), todos los experimentos realizados muestran valores muy parecidos, si bien destacar que únicamente la hibridación “MOPSO-8000 + ESN-17000” consigue superar a “ESN-25000”.

TABLA IX  
HIPERVOLUMEN

	HV
<b>ESN-8000 + MOPSO-17000</b>	0,7127
<b>ESN-12500 + MOPSO-12500</b>	0,7119
<b>ESN-17000 + MOPSO-8000</b>	0,7139
<b>MOPSO-8000 + ESN-17000</b>	0,7167
<b>MOPSO-12500 + ESN-12500</b>	0,7135
<b>MOPSO-17000 + ESN-8000</b>	0,7090
<b>MOPSO-25000</b>	0,7083
<b>ESN-25000</b>	0,7149

En la Tabla X se muestra la dominancia que cada una de las hibridaciones experimentadas obtienen sobre las soluciones que MOPSO o ESN consiguen para este problema. En las Tablas XI y XII se observa el número de puntos que los frentes finales, resultado de las ejecuciones con MOPSO y ESN de manera individual, dominan sobre las soluciones obtenidas por las diferentes hibridaciones probadas.

Como punto de partida se analiza el valor del SC resultante de la ejecución en la que se usa, inicialmente, el algoritmo de ESN dando paso a continuación al algoritmo de MOPSO. Si se compara el porcentaje de dominancia de estas tres primeras hibridaciones sobre MOPSO en solitario, con el SetCoverage obtenido por éste último sobre ellas se puede ver que, en los tres casos, la hibridación ha conseguido superar el valor métrico logrado por las soluciones que “MOPSO-25000” ha dado para MANETs. Pero no ocurre lo mismo con “ESN-25000” ya que, tomando el SetCoverage como medida, aporta mejores frentes de Pareto (un SC mayor) que las hibridaciones para el problema a



resolver (ver Tabla X y Tabla XI). De esta manera, no se puede decir que la hibridación en la que comienza la ejecución ESN consiga superar a la ejecución de los algoritmos de manera individual.

TABLA X  
SETCOVERAGE

SC	MOPSO-25000	ESN-25000
MOPSO-25000		0,3548
ESN-25000	0,4581	
ESN-8000 + MOPSO-17000	0,431	0,39
ESN-12500 + MOPSO-12500	0,451	0,396
ESN-17000+ MOPSO-8000	0,4115	0,3550
MOPSO-8000 + ESN-17000	0,4465	0,4265
MOPSO-12500 + ESN-12500	0,4885	0,4
MOPSO-17000 + ESN-8000	0,4299	0,3417

TABLA XI  
SETCOVERAGE

SC	ESN-8000 + MOPSO-17000	ESN-12500 + MOPSO-12500	ESN-17000 + MOPSO-8000
MOPSO-25000	0,3975	0,3430	0,3575
ESN-25000	0,4075	0,4200	0,415

Siguiendo con el SC como medida, se observa claramente que las dos primeras hibridaciones en las que comienza la ejecución MOPSO obtienen un valor de SetCoverage mayor. Es decir, cada uno de estos dos experimentos consigue dominar más puntos en los frentes resultado de la ejecución de MOPSO o de ESN en solitario que a la inversa (ver Tablas X y XII). Como se puede observar, el SetCoverage de “MOPSO-8000 + ESN-17000” sobre “ESN-25000” (42,65% de puntos dominados) es mayor que el SC de éste último sobre la hibridación (37,9%), lo mismo ocurre con “MOPSO-12500 + ESN-12500”. El comportamiento de estos dos experimentos para con “MOPSO-25000” es similar al anterior, sus frentes consiguen mayor porcentaje de dominancia que las soluciones aportadas por MOPSO en solitario. Si bien parece ser que, a medida que se aumenta el número de evaluaciones en MOPSO y se disminuyen de ESN (hasta que ambos quedan igualados), los frentes de Pareto de las soluciones parecen mejorar a los obtenidos por cada uno de los algoritmos por separado. Pero, al superar MOPSO a ESN en

evaluaciones, estos frentes empeoran ya que ESN precisa de más iteraciones para conseguir cubrir las deficiencias de MOPSO.

TABLA XI  
SETCOVERAGE

SC	MOPSO-8000 + ESN-17000	MOPSO-12500 + ESN-12500	MOPSO-17000 + ESN-8000
MOPSO-25000	0,3283	0,3575	0,3633
ESN-25000	0,379	0,3800	0,4883

Si se tiene en cuenta también la medida de Hipervolumen, se puede decir que la hibridación que mejor ha funcionado en la resolución de MANETs es aquella en la que comienza la ejecución MOPSO usando 8000 evaluaciones dando paso a ESN el cual ejecuta 17000 evaluaciones. Este tipo de combinación consigue superar tanto el HV y el SC de las soluciones aportadas por la ejecución de MOPSO con 25000 evaluaciones y ESN con 25000 evaluaciones.

Una vez comparado el comportamiento de las hibridaciones de los algoritmos, se puede afirmar que el tipo de hibridación propuesto permite obtener frentes de Pareto mejores que los obtenidos individualmente por dichos algoritmos para la resolución de MANETs.

## V. CONCLUSIONES Y LÍNEAS FUTURAS

El objetivo de este trabajo de investigación ha sido estudiar la utilidad del uso de la hibridación entre dos algoritmos multiobjetivo: Enjambres de partículas (MOPSO) y un algoritmo multiobjetivo basado en la combinación de NSGA-II con Estrategias Evolutivas (ESN). Ésta consistía en utilizar la población obtenida por un algoritmo para inicializar el otro. Se pretendía analizar si dicha hibridación permitía obtener frentes de Pareto mejores que los obtenidos individualmente por los algoritmos. Para ello, se ha llevado a cabo una experimentación con cuatro funciones teóricas y un problema real. Los frentes obtenidos a partir de ésta han sido evaluados con las métricas de Hipervolumen y SetCoverage.

En las pruebas realizadas con las cuatro funciones teóricas, se observó que la hibridación no es de utilidad para los problemas ZDT1 y ZDT3. Sin embargo, en ZDT2 y ZDT4, sí aporta soluciones que consiguen mejorar notablemente los frentes obtenidos por la ejecución de MOPSO o ESN en solitario. Para la ZDT2 se usó ESN con 8000 evaluaciones más MOPSO con 17000, mientras que para la ZDT4 funcionó mejor la hibridación de ESN con 12500 evaluaciones más MOPSO con otras 12500. Se piensa que ESN sirve para iniciar el algoritmo, localizando individuos aptos para el

frente de Pareto, dando paso a MOPSO el cual se ocupa de refinarlo y mejorarlo encontrando aquellos puntos que sean más dominantes. De esta manera, se evita que MOPSO se atasque en el inicio del algoritmo.

En el caso de MANETs también se demostró que la hibridación puede ayudar a mejorar los frentes obtenidos de las ejecuciones simples de MOPSO o ESN. Más concretamente se ha observado que la combinación de primero MOPSO con 8000 evaluaciones y después ESN con 17000 es la que consigue mayores valores métricos en sus frentes. Se piensa que esto es debido a que MOPSO ayuda a mejorar la dispersión del frente en los primeros ciclos de algoritmos mientras que ESN afina más dicho frente en posteriores iteraciones.

En función de lo anterior, queda probada la utilidad del uso del tipo de hibridación propuesto para este artículo ya que esta técnica es capaz de conseguir mejores soluciones que las alcanzadas hasta el momento para determinados problemas. Esto se debe a que un algoritmo ayuda a subsanar las deficiencias del otro.

Como trabajo futuro de investigación se propone estudiar la utilidad de otro tipo de hibridación. Ésta consistiría en migrar o intercambiar individuos entre poblaciones de los dos algoritmos durante la ejecución.

#### AGRADECIMIENTOS

Este trabajo ha sido financiado por el proyecto de investigación OPLINK::UC3M, Ref: TIN2005-08818-C04-02.

#### REFERENCIAS

- [1] E. Alba, B. Dorronsoro, F. Luna, A.J. Nebro, P. Bouvry, L. Hogie, A Cellular Multi-Objective Genetic Algorithm for Optimal Broadcasting Strategy in Metropolitan MANETs, *Computer Communications*, 30(4):685-697, 2007.
- [2] S. Bleuler, M. Laumanns, L. Thiele y E. Zitzler: PISA—a platform and programming language independent interface for search algorithms. *Evolutionary Multi-Criterion Optimization (EMO 2003)*, 2003, pp. 494-508.
- [3] C. Coello Coello y M. Reyes-Sierra: Fitness Inheritance In Multi-Objective Particle Swarm Optimization. En 8th annual conference on Genetic and evolutionary computation, Seattle, Washington (USA), 2006.
- [4] C. Coello Coello: MOPSO: a proposal for multiple Objective Particle Swarm Optimization. *Proceedings of the Congress on Evolutionary Computation*, 2002.
- [5] K. Deb, S. Agrawal, A. Pratap y T. Meyarivan: A Fast and Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, 2000, pp. 849-858.
- [6] L. Hogie, F. Guinand y P. Bouvry: A heuristic for efficient broadcasting in the metropolitan ad hoc network. En 8th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems, 2004, pp. 727-733.
- [7] J.M Fernández, G. Jiangreco y Benjamín Barán: Equipo Probabilístico de Algoritmos Evolutivos Multiobjetivo en Computación Paralela. En 7º Simposio Argentino de Inteligencia Artificial - ASAI2005 Rosario, 29-30 de Agosto de 2005.
- [8] J. D. Knowles y D. W. Corne: Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 2000, pp. 149-172.
- [9] L. Hogie, F. Guinand y P. Bouvry: The madhoc metropolitan adhoc network simulator. Technical report, Université du Luxembourg and Université du Havre, Francia, marzo 2006. <http://www.lih.univlehavre.fr/hogie/madhoc/>.
- [10] C. León, G. Miranda y C. Segura: Herramienta paralela para optimización multiobjetivo basada en un modelo de islas auto-adaptativo. Guía de usuario. Versión 0.5. Universidad de la Laguna, 2007.
- [11] F. Luna, A.J. Nebro, B. Dorronsoro, E. Alba, P. Bouvry, and L. Hogie. Optimal broadcasting in metropolitan manets using multiobjective scatter search. In *EvoWorkshops*, pages 255-266, 2006.
- [12] R. Pérez Pérez, C. Luque, A. Cervantes y P. Isasi: Multiobjective Algorithms to Optimize Broadcasting Parameters in Mobile Ad-hoc Networks. *IEEE Congress on Evolutionary Computation*, 2007, pp. 3142-3149.
- [13] E-G. Talbi: A Taxonomy of Hybrid Metaheuristics. *Journal of Combinatorial Optimization*. Kluwer Academic Publishers, Boston. 1999, pp. 1-45.
- [14] E. Talbi, S. Cahon y N. Melab: ParadisEO: A Framework for the Reusable Design of Parallel and Distributed Metaheuristics. *Journal of Heuristics*, 2004, pp. 357-380.
- [15] E. Zitzler, K. Deb, y L. Thiele. Comparison of Multiobjective Evolutionary Algorithms on Test Functions of Different Difficulty. En A. S. Wu, editor, *Proceedings of the 1999 Genetic and Evolutionary Computation Conference. Workshop Program*, Orlando, Florida, 1999, pp. 121-122.
- [16] E. Zitzler, M. Laumanns y L. Thiele: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. *EUROGEN*, 2001, pp. 95-100.